

```

1 <?php
2
3 /**
4  * @author GOLAY Brian
5  * @version 1.0 (2021/05/20)
6  * Add flight logic here
7  */
8
9 //require all the flight functions
10 require_once("../sql/flightDAO.php");
11 //require all the media functions
12 require_once("../sql/mediaDAO.php");
13 //require all the database functions
14 require_once("../sql/dbConnection.php");
15
16 use FlightClub\sql\FlightDAO;
17 use FlightClub\sql\MediaDAO;
18 use FlightClub\sql\DBConnection;
19
20 //check if the user is not already connected
21 if (!isset($_SESSION['userID'])) {
22     header("Location: ../index.php?page=homepage");
23     exit();
24 }
25
26 //we hide error to the user to only show our custom errors
27 error_reporting(0);
28
29 //we filter the user input
30 $role = filter_input(INPUT_POST, "Cd_Role", FILTER_SANITIZE_STRING);
31 $flight = filter_input(INPUT_POST, "No_Flight", FILTER_SANITIZE_STRING);
32 $dateDeparture = filter_input(INPUT_POST, "Dttm_Departure", FILTER_SANITIZE_STRING);
33 $dateArrival = filter_input(INPUT_POST, "Dttm_Arrival", FILTER_SANITIZE_STRING);
34 $timeEngineOn = filter_input(INPUT_POST, "Tm_Engine_On", FILTER_SANITIZE_STRING);
35 $timeEngineOff = filter_input(INPUT_POST, "Tm_Engine_Off", FILTER_SANITIZE_STRING);
36 $typeAircraft = filter_input(INPUT_POST, "Nm_Plane", FILTER_SANITIZE_STRING);
37 $registrationPlane = filter_input(INPUT_POST, "No_Plane", FILTER_SANITIZE_STRING);
38 $icaoDeparture = filter_input(INPUT_POST, "Cd_ICAO_Departure",
    FILTER_SANITIZE_STRING);
39 $icaoArrival = filter_input(INPUT_POST, "Cd_ICAO_Arrival", FILTER_SANITIZE_STRING);
40 $flightType = filter_input(INPUT_POST, "Cd_Flight_Type", FILTER_SANITIZE_STRING);
41 $flightMode = filter_input(INPUT_POST, "Cd_Flight_Mode", FILTER_SANITIZE_STRING);
42 $weather = filter_input(INPUT_POST, "Txt_Meteo", FILTER_SANITIZE_STRING);
43 $passengers = filter_input(INPUT_POST, "Nb_Passengers", FILTER_SANITIZE_NUMBER_INT);
44 $notes = filter_input(INPUT_POST, "Txt_Note", FILTER_SANITIZE_STRING);
45 $btn = filter_input(INPUT_POST, 'AddFlight', FILTER_SANITIZE_STRING);
46 $allflight = "";
47 $validTime = true;
48 $engineTime = true;
49 $idFlight = 0;
50 $error = "";
51
52 /**
53  * Function found on Stack overflow to check if a time is valid. Found at:
54  * https://stackoverflow.com/questions/3964972/validate-that-input-is-in-this-time-format-hhmm
55  *
56  * @param string $time
57  * @return bool
58  */

```

```

58 function checkTime($time)
59 {
60     //check that the first char is either 0, a 1 or a 2, then the second character
    between 0 and 9. Then it looks for the ":" character. Then it check that the first
    minute number is between 0 and 5, finnaly, it check that the last number is between 0
    and 9
61     if (preg_match("/^(?:2[0-3]|[01][0-9]):[0-5][0-9]$/", $time)) {
62         return true;
63     } else {
64         return false;
65     }
66 }
67
68 /**
69  * function that validate the date
70  *
71  * @param string $date
72  * @return bool
73  */
74 function validateDate($date)
75 {
76     $result = "";
77     $newDate = explode("T", $date);
78     try {
79         $result = checkdate(explode("-", $newDate[0])[1], explode("-", $newDate[0])
[2], explode("-", $newDate[0])[0]);
80     } catch (\Throwable $th) {
81         $result = null;
82     }
83     return $result;
84 }
85
86 //we reset the message
87 $message = array();
88 $successMessage = "";
89 //if all data are not empty
90 if (!empty($btn)) {
91
92     if (
93         !empty($role) && !empty($flight) && !empty($dateDeparture) &&
!empty($dateArrival) &&
94         !empty($typeAircraft) && !empty($registrationPlane) && !empty($icaoDeparture)
&& !empty($icaoArrival) && !empty($flightType)
95         && !empty($flightMode) && !empty($weather) && $passengers != ""
96     ) {
97     } else {
98         $message[] = "Entrez tout les champs";
99     }
100     //we check that the data is valid
101     if (strlen($icaoDeparture) == 4 && strlen($icaoArrival) == 4) {
102     } else {
103         $message[] = "Vérifiez les codes ICAO";
104     }
105     //check that dates are valid
106     try {
107         if (validateDate($dateDeparture) && validateDate($dateArrival)) {
108         } else {
109             $message[] = "La date n'est pas valide";
110         }
111     } catch (\Throwable $th) {

```

```
112     $message[] = "La date n'est pas valide";
113 }
114
115 //check the validity of the hours of the take off time and landing time
116 try {
117     if (checkTime(explode("T", $dateDeparture)[1]) && checkTime(explode("T",
118 $dateArrival)[1])) {
119         $validTime = true;
120         //check the validity of the hours for the engine off and engine on time
121         if (!empty($timeEngineOn) && !empty($timeEngineOff)) {
122             $engineTime = true;
123             if (checkTime($timeEngineOn) && checkTime($timeEngineOff)) {
124                 $validTime = true;
125             } else {
126                 $validTime = false;
127             }
128         } else {
129             $timeEngineOff = null;
130             $timeEngineOn = null;
131         }
132     } else {
133         $message[] = "La date et l'heure saisie ne sont pas valides";
134     }
135 } catch (\Throwable $th) {
136     $message[] = "L'heure saisie n'est pas valide";
137 }
138
139 if ($validTime && $engineTime) {
140     //check that the type of aircraft doesn't blow the database
141 } else {
142     $message[] = "L'heure saisie n'est pas valide";
143 }
144 if (strlen($typeAircraft) <= 45) {
145     //check that the registration of the aircraft doesn't blow the database
146 } else {
147     $message[] = "Le type d'avion est trop long";
148 }
149 if (strlen($registrationPlane) <= 45) {
150     //check that the flight type doesn't blow the database
151 } else {
152     $message[] = "L'immatriculation de l'avion est trop longue";
153 }
154 if (strlen($flightType) <= 45) {
155     //check that the flight mode doesn't blow the database
156 } else {
157     $message[] = "Le type de vol est trop long";
158 }
159 if (strlen($flightMode) <= 45) {
160     //check that the weather doesn't blow the database
161 } else {
162     $message[] = "Le mode de vol est trop long";
163 }
164 if (strlen($weather) <= 512) {
165     //check that the passengers number doesn't blow the database
166 } else {
167     $message[] = "La météo est trop longue";
168 }
169 if ($passengers <= 9999999999 && $passengers >= 0) {
170     //check that the notes doesn't blow the database
171 } else {
```

```
171     $message[] = "Le nombre de passagers est invalide";
172 }
173 if (strlen($notes) <= 1024) {
174     //every data is correct
175 } else {
176     $message[] = "Les notes sont trop longues";
177 }
178
179 //we check that every data is ok
180 if (empty($message)) {
181
182     //we check with is the latest id in the database
183     $allflight = FlightDAO::getAllFlightOrderById();
184     $idFlight = 0;
185     if (!empty($allflight)) {
186         $idFlight = $allflight[0]['Id_Flight'] + 1;
187     } else {
188         $idFlight = 0;
189     }
190     try {
191
192         //converting departure datetime to a good datetime
193         $dateDeparture = explode("T", $dateDeparture);
194         $dateDeparture = $dateDeparture[0] . " " . $dateDeparture[1];
195
196         //converting arrival datetime to a good datetime
197         $dateArrival = explode("T", $dateArrival);
198         $dateArrival = $dateArrival[0] . " " . $dateArrival[1];
199
200         //we add the flight
201         FlightDAO::addFlight(
202             $idFlight,
203             $_SESSION['userID'],
204             $role,
205             strtoupper($flight),
206             $dateDeparture,
207             $dateArrival,
208             $timeEngineOn,
209             $timeEngineOff,
210             $typeAircraft,
211             strtoupper($registrationPlane),
212             strtoupper($icaoDeparture),
213             strtoupper($icaoArrival),
214             $flightType,
215             $flightMode,
216             $weather,
217             $passengers,
218             $notes
219         );
220         //we show a success message
221         $successMessage = "Vol enregistré avec succès!";
222     } catch (\Throwable $th) { //we show an error message
223         $message = "Une erreur est survenue. Merci de réessayer";
224     }
225     /////FILE UPLOAD
226
227     $name = "";
228     $lienimg = "";
229     $default_dir = "./media/";
230     $errorimg = "";
```

```

231     $size_total = 0;
232     $error = "";
233     $type = '';
234     $extension = '';
235     $nb_files = count($_FILES['media']['name']);
236     $MAX_FILE_SIZE = 3145728; // 3MB in bytes
237     $MAX_POST_SIZE = 73400320; // 70MB in bytes
238
239     $extensions = array(
240         "image" => array('.png', '.gif', '.jpg', '.jpeg'),
241     );
242     // available types of file
243     $types = array('image');
244     if ($_FILES['media']['name'][0] != "") {
245         DBConnection::startTransaction();
246         //we check the size of the image
247         foreach ($_FILES['media']['size'] as $key => $value) {
248             if ($value > $MAX_FILE_SIZE) {
249                 $error = 'File too heavy.';
250                 DBConnection::rollback();
251             } else {
252                 $size_total += $value;
253             }
254         }
255         //if there is images we check each media size
256         if (isset($_FILES['media'])) {
257             for ($i = 0; $i < $nb_files; $i++) {
258                 $errorimg = $_FILES['media']['error'][$i];
259                 if ($error == 'File too heavy.' || $size_total >
260 $MAX_POST_SIZE) {
261                     $error = "Fichier trop volumineux!";
262                     DBConnection::rollback();
263                 } else {
264                     //we check the mime type
265                     $separator = '/';
266                     $extension = explode($separator,
mime_content_type($_FILES['media']['tmp_name'][$i]))[1];
267                     $type = explode($separator,
mime_content_type($_FILES['media']['tmp_name'][$i]))[0];
268
269                     if (!in_array($type, $types)) {
270                         $error = "erreur dans le type de fichier";
271                         DBConnection::rollback();
272                     } else {
273                         if ($error != "erreur dans le type de fichier") {
274                             if ($errorimg[0] == 0) { //if there is no error,
we add the file
275                                 $tmp_name = $_FILES['media']['name'][$i];
276                                 $name = explode(".", $tmp_name);
277                                 $name = $name[0] . uniqid() . "." . $name[1];
278
279                                 if (move_uploaded_file($_FILES['media']
["tmp_name"][$i], $default_dir . $type . "/" . $name)) {
280                                     //add the filename in the database
281                                     MediaDAO::changePath($name,
$tmp_name[$i]);
282                                     $lienimg = $default_dir . $type . "/" .
$name;
283                                 } else {

```

```
284         DBConnection::rollback();
285     }
286 }
287 try {
288     MediaDAO::addmedia($lienimg, $idFlight);
289
290     DBConnection::commit();
291 } catch (\Throwable $th) {
292     $error = $th;
293     DBConnection::rollback();
294 }
295 } else {
296     DBConnection::rollback();
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 if ($error != "") {
306     $message .= " " . $error;
307 }
308
```